Modelling, Design and Control of a Four wheel Holonomic Drive

Paras Savnani, Harshvardhan Sisodia, Divyanshu Tak and Akash Mecwan

December 1, 2019

Modelling, Design and Control of a Four wheel Holonomic Drive

Paras S Savnani, Mechanical Branch, Institute of technology, Nirma University, Ahmedabad,India savnani5@gmail.com

Harshvardhan S Sisodia, Mechanical Branch, Institute of technology, Nirma University, Ahmedabad, India. 16bme067@nirmauni.ac.in

Divaynshu Tak, Electronics and Communication Branch, Institute of technology, Nirma University, Ahmedabad, India. 16bec037@nirmauni.ac.in

Dr. Akash Mecwan, Electronics and Communication Branch, Institute of technology, Nirma University, Ahmedabad, India. Akash.mecwan@nirmauni.ac.in

***Abstract*—This paper tries to model a four wheel holonomic drive and it describes the design of the drive system. After deriving the kinematic model, the encoder and the gyroscope equations are used for position and orientation feedbacks respectively. This drive is designed using various selection criterion for microcontroller, motors, motor drivers and sensors. Furthermore, it is controlled using deduced reckoning algorithm and a continuously running PID is applied to keep the errors in check. Also, the ideal velocity vs time curve is discussed in the paper to achieve the desired outcome in minimum possible time.**

***Keywords*—Deduced Reckoning, Holonomic drive, Kinematic model, PID controller, Rotary Encoders**

I. INTRODUCTION

Mobile robots have different modes of locomotion and among them wheeled robots are most commonly used on smooth and flat surfaces. Mainly, wheeled robots consist of differential drives, Ackerman steering [3], omni-drive and synchro-drive robots. All the systems have certain advantages and disadvantages in terms of controllability and maneuverability.

The omni-drive system is superior to other drives because it has better dexterity and driving ability. This drive can move in any direction without altering its orientation. But, there is an inverse correlation between controllability and maneuverability. Controlling an omnidirectional robot for a specific direction of travel is difficult and often less accurate when compared to less maneuverable designs [4]. Therefore, odometric sensors are used for feedback with this drive to minimize the errors in the robot's trajectory. Moreover, owing to its superiority, this drive has widespread applications ranging from warehouse automation to on road automobiles.

This paper presents the kinematic model to control the omni-drive and proposes the hardware selection criterion along with control algorithms. The presented algorithms combined with the hardware were experimentally tested and verified on an autonomous mobile robot platform and the results were very accurate with errors in the tolerance limit.

II. KINEMATIC MODEL

Figure 1 shows the top view of the robot in global coordinate system and to represent the robot position, the relation between local (X_L, Y_L) and global coordinate systems (X_G, Y_G) is established. The point O on robot chassis is represented by (X, Y) and the angular difference between the frames is given by θ. The position of robot is defined as the vector of these three elements.

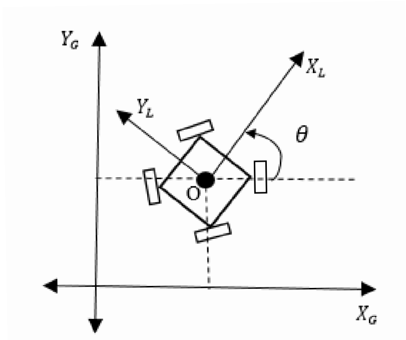

Fig 1. Comparison of robot's local and global reference frames

$$P_G = \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} \quad (1)$$

To represent the robot motion, it is required to map motion along global axes to robot's local axes, thus the rotational matrix is used for the mapping purpose.

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The mapping is the function of the current position of the robot and the operation is denoted by:

$$\dot{P}_L = R(\theta)\dot{P}_G \quad (3)$$

Here $\dot{P}_L$ denotes the velocities in the local frame and $\dot{P}_G$ denotes the velocities in the global frame.

As, the omni drive has four wheels, so the robot's motion in global reference frame can be generated by computing the contribution of each wheel in the local reference frame. Now, the omni wheel is a special type of wheel which has an extra degree of freedom over the standard wheel. It consists of rollers attached to the periphery of the wheel which allows it to move omnidirectionally. Figure 2 shows the omni wheel in the local reference frame and it lists the parameters which can be controlled. Here, l depicts the distance of the wheel from the robots centre point O. Angle λ depicts the angular offset from the horizontal axis, angle α shows the angle of wheel plane relative to the chassis and finally angle ψ shows the angle of rollers with centre wheel plane. Also, r is the radius of the wheel and φ is the rotation of the wheel.

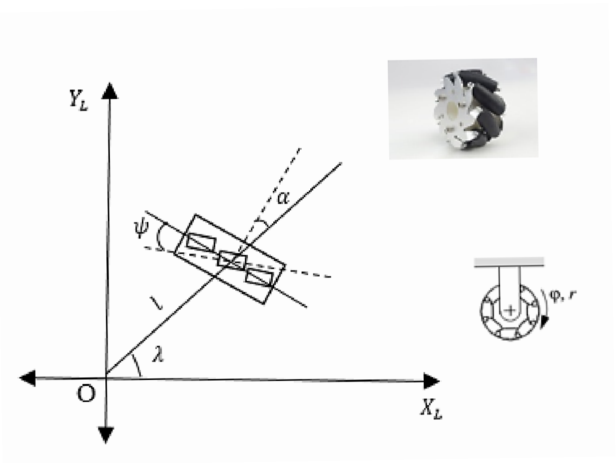

Fig 2. Omni wheel parameters

The rolling constraint for the wheel enforces that the motion along the wheel plane should be accompanied by correct amount of wheel spin so as to ensure pure rolling at contact point.

$$[\ \sin(\lambda+\alpha+\psi) \quad -\cos(\lambda+\alpha+\psi) \quad -l\cos(\alpha+\psi)] * R(\theta)\dot{P}_G = r\dot{\varphi}\cos\psi \tag{4}$$

The first term of the sum shows the total motion along wheel plane. The three elements of the vector represent mappings from each of $\dot{x}, \dot{y}, \dot{\theta}$ to their contribution of motion along the wheel plane and $R(\theta)\dot{P}_G$ is used to transform the motion parameters from global to local frame. It is essential because all the parameters, l, λ, α, ψ are in robot's local frame. Thus, the motion along the wheel plane must be equal, according to this constraint, to the motion accomplished by spinning the wheel i.e. $r\dot{\varphi}\cos\psi$. Also, motion orthogonal to this direction is not constrained due to the freely rotating rollers having angular velocity $\dot{\varphi}_r$ and radius r_r .

$$[\cos(\lambda+\alpha+\psi) \quad \sin(\lambda+\alpha+\psi) \quad l\sin(\alpha+\psi)] * R(\theta)\dot{P}_G = r\dot{\varphi}\sin\psi + r_r\dot{\varphi}_r \tag{5}$$

To formulate the kinematic constraints of the omni drive robot, the kinematic constraints of each wheel are combined based on their relative placement in the robot. Here, omni wheels with $\psi = 0$ are used thus the rollers have no angle with respect to the centre wheel plane. Due to this, the motor motion only contributes to rolling constraints in wheel plane and the wheel is free to move in the direction perpendicular to the wheel plane. So, we will not consider equation 5 in our calculations.

Figure 3 represents the actual structure of the robot and the conventions. Firstly, a local reference frame is attributed to the robot and the robot's centre is the origin (O) for the frame with each wheel at 90° with respect to each other. The heading of the robot is taken in the direction of X_L and each wheel has a certain velocity i.e. $v = r\dot{\varphi}$.

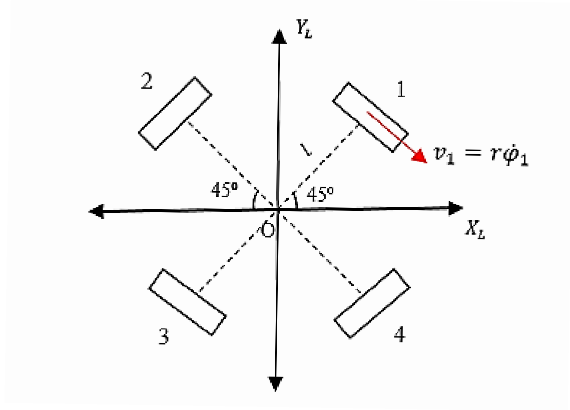

Fig 3. Robot Convention

Collecting the rolling constraints of each wheel in a single expression yields the equation:

$$F_1 R(\theta)\dot{P}_G = F_2 \tag{6}$$

Where, F_1 is the combined matrix of rolling constraints of each wheel and F_2 is the combined velocity matrix of each wheel. The equation 6 can be further written as:

$$\begin{bmatrix} \sin(\lambda_1+\alpha+\psi) & -\cos(\lambda_1+\alpha+\psi) & -l\cos(\alpha+\psi) \\ \sin(\lambda_2+\alpha+\psi) & -\cos(\lambda_2+\alpha+\psi) & -l\cos(\alpha+\psi) \\ \sin(\lambda_3+\alpha+\psi) & -\cos(\lambda_3+\alpha+\psi) & -l\cos(\alpha+\psi) \\ \sin(\lambda_4+\alpha+\psi) & -\cos(\lambda_4+\alpha+\psi) & -l\cos(\alpha+\psi) \end{bmatrix} * \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r\dot{\varphi}_1\cos\psi \\ r\dot{\varphi}_2\cos\psi \\ r\dot{\varphi}_3\cos\psi \\ r\dot{\varphi}_4\cos\psi \end{bmatrix} \tag{7}$$

From figure 3, substituting the values $\alpha = 0$, $\psi = 0$, $\lambda_1 = 45^0$, $\lambda_2 = 135^0$, $\lambda_3 = 225^0$ and $\lambda_4 = 315^0$ in equation 7 and solving for velocities:

$$v_1 = \frac{\dot{X}}{\sqrt{2}}(\cos\theta + \sin\theta) + \frac{\dot{Y}}{\sqrt{2}}(\sin\theta - \cos\theta) - l\dot{\theta} \tag{8}$$

$$v_2 = \frac{\dot{X}}{\sqrt{2}}(\cos\theta - \sin\theta) + \frac{\dot{Y}}{\sqrt{2}}(\sin\theta + \cos\theta) - l\dot{\theta} \tag{9}$$

$$v_3 = \frac{\dot{X}}{\sqrt{2}}(-\cos\theta - \sin\theta) + \frac{\dot{Y}}{\sqrt{2}}(-\sin\theta + \cos\theta) - l\dot{\theta} \tag{10}$$

$$v_4 = \frac{\dot{X}}{\sqrt{2}}(-\cos\theta + \sin\theta) + \frac{\dot{Y}}{\sqrt{2}}(-\sin\theta - \cos\theta) - l\dot{\theta} \tag{11}$$

To track the position of the robot in global reference frame certain sensors are required to calculate the distance in both axes, and it is very difficult to calculate the distance using laser or other non-contact based sensors so we opt for optical rotary encoders. Although, they have incremental errors but those are overcome by it by its advantage i.e. it can help the robot to navigate without knowing the environment beforehand. Initially, encoders were coupled with motors but this configuration yielded inaccurate results due to wheel slippage so, the L configuration shown in figure 4 was used, where each encoders are spring suspended and are aligned with the local reference frame of the robot.

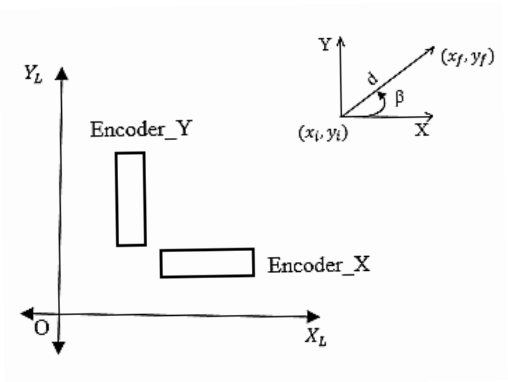

Fig 4. Encoder mounting with respect to local reference frame

Now, to track the coordinates of the robot, let's assume the number of pulses given by Encoder_X as $Pulses_X$ and the number of pulses given by Encoder_Y as $Pulses_Y$ and the distance traveled by the robot per pulse is $\frac{2\pi r}{Pulses_{rev}}$, where r is the radius of wheel coupled to encoder and $Pulses_{rev}$ are the pulses per revolution.

The distance traveled by each encoder is given by:

$$\text{Encoder\_X} = Pulses_X * \frac{2\pi r}{Pulses_{rev}} \tag{12}$$

$$\text{Encoder\_Y} = Pulses_Y * \frac{2\pi r}{Pulses_{rev}} \tag{13}$$

Thus the total distance travelled by the robot (d) =

$$\sqrt{(Pulses_X * \frac{2\pi r}{Pulses_{rev}})^2 + (Pulses_Y * \frac{2\pi r}{Pulses_{rev}})^2} \tag{14}$$

Assuming the initial coordinates of the robot to be (x_i, y_i) and the final coordinates to be (x_f, y_f), then the heading (β) of the robot is:

$$\beta = \tan^{-1}\left(\frac{y_f - y_i}{x_f - x_i}\right) \tag{15}$$

Angular feedbacks are necessary to maintain the orientation of the robot while moving and it is achieved using a gyroscope. If the orientation is required to be locked at an angle θ_f and the gyroscope detects a change in angle $d\theta$, the new angle (θ_f) is calculated by adding the difference $d\theta$ to θ_i (current angle).

$$\theta_f = \theta_i + d\theta \tag{16}$$

III. Design of drive

A. Mechanical configuration

The mechanical frame supporting the drive is made up of aluminium extrusions and it has custom built motor mounts for mounting the motors. Additionally as it is a four wheel drive, there is a requirement of a suspension system to maintain ground contact at all wheels in the drive, so each motor is spring suspended to maintain the contact at all times. Figure 5 shows the actual system where the wheels are each 90º apart and their rotation axis is radial with respect to the centre of robot. Also, to increase the positional accuracy electromechanical brakes are used with each wheel to improve the breaking. Furthermore, suspended encoders are used in L configuration as shown in figure to receive the XY coordinates of the system. Apart from this, the yaw rate gyro sensor is mounted on the flat base parallel to the ground and at the centre of the robot to minimize the vibrations. The wheels used are 6" omni wheels [5], which have a special construction i.e. they do not restrain movement in the direction parallel to the rotation axis and they are directly coupled with the gearbox's shaft.

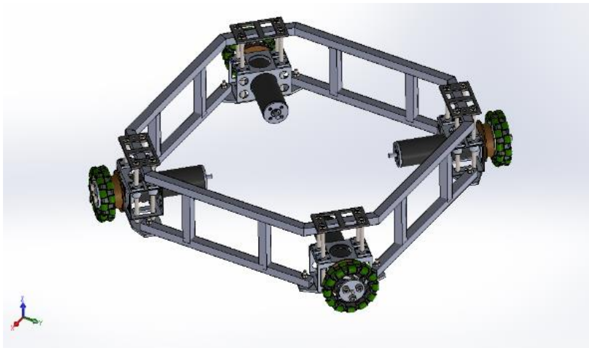

Fig 5. Software Model of four wheel omni drive

B. Motor Selection

The motors are used to drive the system and they should be chosen considering factors like the load on the drive (torque requirements), speed of operation, type of terrain, voltage limitations, type of wheels, weight and mounting of motors etc. Also the maximum allowable current of the motor driver should be taken into consideration. For this application, 200W brushed DC Maxon^{TM} motors (Model no: 618570) [6] with a planetary gear box (Model no: 223081) with reduction 4.3:1 are used. This motor provides high torque and high speed operation in harsh and unfavorable environmental conditions due to its sturdy design.

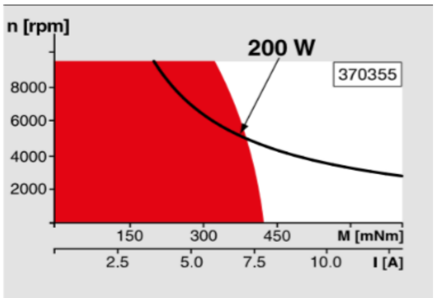

Fig 6. Motor characteristics [6]

C. *Microcontroller Selection*

In this system, the microcontroller controls the system by executing the commands and issuing them to the motor driver unit. It is also responsible for the calculations and processing required to run the drive. The controller must be chosen to cater the needs of our system and it must have desired clock speed, RAM, I/O pins, sufficient ports etc. For instance, if the feedback of the angle is taken using SPI (Serial Peripheral Interface) protocol then the controller must have inbuilt SPI port. If the motor controller accepts commands using some standard communication protocol, then the hardware required for communication is also required to be taken into consideration. For this application a microcontroller powered by a high performance ARM® Cortex® M3 processor LPC 1769 [7] running at 120 MHz is used. High speed is essential for fast floating point math operations, required for the generation of machine coordinates.

D. *Motor Driver*

A motor driver is a circuit used to run the motor at desired RPM and in the desired direction. A DC motor driver is able to run the motor in both directions and provides a large variety of intermediate speeds between zero and maximum RPM of the motor. The driver works on the principle of PWM (Pulse Width Modulation) signal being fed to MOSFETS to drive the motors with variable speeds and H-bridge principle for direction reversal. PWM signals with high resolution are better for smooth control as they make more interpolations possible between the RPM range of the motor providing superior control with greater accuracy. The motor driver should be chosen according to the current rating of the motors to be used, also drivers with inbuilt controller should be preferred to drive more motors to reduce the load on master controller. For this application, Sabertooth Dual Channel 32A [8] motor drivers are used with 24V input voltage.

E. *Odometric Sensors for feedback*

a) Optical rotary encoders:

Optical rotary encoders are used to determine the real time position of the robot in the global coordinate system. There are two types of encoder i.e. absolute and incremental. Here, incremental encoders are used because of their cheap rate and fast data transfer rates than absolute shaft encoders. For our application we use Pepperl+Fuchs Optical Incremental Encoders (Model no: TVI40N) [9] with 1024 PPR (Pulses Per Revolution) for high resolution and better positioning precision.

b) Gyroscope sensor:

Gyroscope provides the real time feedback in angular errors in the orientation of the drive system. The use of gyroscope has proved to be very useful due to its accurate output and relatively error free transfer of data. The gyroscope or accelerometer chosen for angle updation must be accurate and its rate of updation should be as fast as possible to avoid error in driving. A high performance and vibration immune yaw rate gyroscope (Model no: ADXRS453) [10] was used in this application. The gyro supports measurement of maximum of 300 deg/s, a high vibration rejection of 0.01 deg/sec/g and has null bias stability of 16 deg/hr.

IV. CONTROL ALGORITHM

A. *Path Planning*

The Motion of omni directional drive can be performed either in straight line path or any curved path. The straight line path is performed by providing heading (15) and feedback distance (12), (13). Due to dynamic errors the robot may not follow the straight line path so, to counter this situation the current coordinates of the robot are updated continuously, keeping the final coordinates (x_f, y_f) same, thus updating the heading (β) continuously, this way the drive remains directed towards the destination.

The curved path can be realized by quantizing a small arc into set of coordinates and subsequently the path can be represented as the sequential combination of the straight lines from those Quantized coordinates. Another method to move the drive on a curved path is by breaking the curve into small arcs of a circle, of given radius R and centre coordinates (x_c, y_c). Here, the heading angle is determined by the instantaneous slope of the tangent at robot's current coordinates (x, y) on the arc using the arctan function. Also, to counter the dynamic errors a perpetually running PID algorithm is implemented on the path and orientation of the robot. Thus, motion in more complex curves and arcs can be performed by sequentially transitioning from one arc to another.

B. *PID Controller*

PID is a control algorithm used to monitor the trajectory of the robot. It is a closed loop controller which minimizes the errors in the trajectory using the feedbacks from the sensors. It calculates the error and applies the correction which includes proportional, integral and derivative elements. The robot can follow any curve to navigate its environment i.e. a straight line, a circle, a parabola or a sine curve etc. Now for the error generation, the instantaneous coordinates of the drive are substituted in the equation of the trajectory curve. Let e (t) represent the error:

For a straight line path:
$$e(t) = mx + c - y \quad (17)$$

Similarly for a parabola:
$$e(t) = ax^2 + bx + c - y \quad (18)$$

Error for a generalized curve:
$$e(t) = f(x) - y \quad (19)$$

Where, (x, y) are the instantaneous robot coordinates

PID is applied independently on the path as well as the orientation of the robot via values received from the rotary encoders and the gyro sensor. Following equation provides the speed control element for the robot:

$$Control\ signal = K_p e(t) + K_d \frac{de(t)}{dx} + K_i \int e(t)dt \quad (20)$$

The time between successive error sampling is decided based on the clock speed of the controller, and is taken as the time between iterations of the calculations. For tuning the PID of the system, machine specific transfer functions were used to model the system with accurate parameters and PID tuning software was used for the precise tuning.

C. Velocity Planning Method

The robot travelling on a defined path in XY plane needs accurate speed and direction inputs for proper path planning. Therefore, the algorithm developed here takes into consideration the motion planning requirements and its target is to complete the distance in minimum possible time with real time control functions.

Let's take the simplified case of the robot travelling on a straight line from a point to another point. The desired outcome is that the robot should reach the destination in minimum possible time. Now, the primary question arises that what velocity vs time curve should it follow to achieve the desired outcome? For linear acceleration and deceleration curves figure 7 illustrates three paths on the curve, which the robot can use i.e. ABFG, ACEG and ADG. Here, an infinite number of paths are possible, but three are shown for explanation purpose. In first two paths the robot moves at a constant velocity for some time. It is evident from the figure that the path ADG has the maximum area under the curve and if the robot accelerates for half the time (or distance) and decelerates for half the time (or distance) it will reach in the shortest time period possible, given that the robot can keep on accelerating without reaching its maximum capacity. Also, it is assumed that the rate of acceleration and deceleration are same (i.e. maximum). If the robot reaches its maximum capacity then it should move at constant velocity and should decelerate for the same time as for acceleration. This result can be generalized for any function be it linear, polynomial or exponential etc.

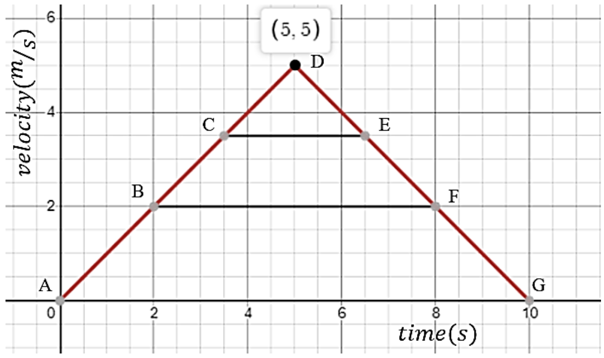

Fig 7. Velocity vs time graph with linear acceleration and deceleration

Now, extrapolating the above generalization for actual system, a step input signal is given to the motors for speed 5 m/s at the beginning and then 0 m/s at the midway distance. The open loop response of the system is exponential as shown in Figure 8, so PID control is applied and the values of the constants K_p, K_d and K_i are determined using a PID tuning software to reduce the latency of the motors. The velocity vs time curve shown in the figure yielded the best results to achieve the set target and this can be further improved upon by using the reversing technique or electromechanical brakes to stop the robot.

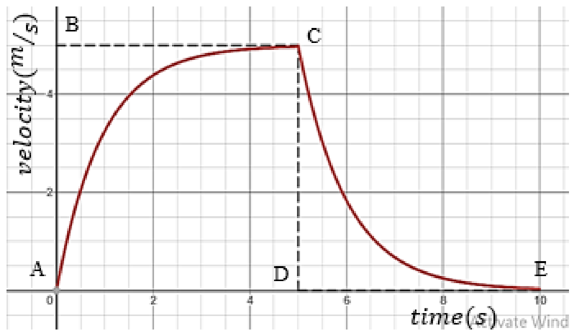

Fig 8. Velocity vs time graph of actual system

V. Conclusion

The kinematic model developed in this paper describes the motion of the four wheel holonomic drive very accurately. Furthermore, the drive design incorporates the robust mechanical structure tested under various conditions and carefully selected microcontrollers, motors and its drivers for rugged usage. This system was tested on wooden flooring with oil paints and it gave positive result with errors in tolerable limit. The proposed PID and speed control algorithms worked very well on the system and can be implemented on other autonomous mobile platforms too. To further improve the system, additional environment data from LIDAR sensors and cameras will help to navigate the path more accurately and will enhance the reliability. Also artificial intelligence search algorithms like A* and bug algorithms can be used to find the optimized path for the robot to reach its destination.

REFERENCES

[1] Jiraphan I and Chirdpong D 2014 “Self-Localization and Navigation of Holonomic Mobile Robot using Omni-Directional Wheel Odometry,” in TENCON 2014 - 2014 IEEE Region 10 Conference.

[2] Vlad B, Ioan M, Viorel G, Mircea N and Monica P 2017 “Encoder-based Path Tracking with Adaptive Cascaded Control for a Three Omni-wheel Robot” IEEE Conferences.

[3] Szakács, Tamás (2010). "Developing Stability Control Theories for Agricultural Transport Systems" (PDF). *Acta polytechnica Hungarica*, pp. 25–37.

[4] R. Siegwart, I.R. Nourbakhsh, “Introduction to Autonomous Mobile Robots, 1st ed. Cambridge, Massachusetts: MIT press, 2004, pp. 47-90.

[5] VexRobotics.com, ‘Vex 6” omni directional wheel with Dual VersaKey Pattern’, [Online]. Available: https://content.vexrobotics.com/vexpro/pdf/217-2613-Drawing20140818.PDF. [Accessed: 14 - July - 2019].

[6] Maxon Motors, “Re 50, ø50, Graphite Brushes, 200W”.[Online]. Available: https://www.maxongroup.com/medias/sys_master/root/8833416560670/19-EN-137.pdf. [Accessed: 20-September-2019]

[7] NXP semiconductors - nxp.com, ‘LPC1769 32-bit ARM Cortex-M3 microcontroller’. [Online]. Available: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0337e/DDI0337E_cortex_m3_r1p1_trm.pdf. [Accessed: 5-October-2019]

[8] Dimensionengineering.com, ‘Sabertooth 2x32 dual channel motor driver’. [Online]. Available: https://www.dimensionengineering.com/datasheets/Sabertooth2x32.pdf. [Accessed: 23-October-2019]

[9] Pepperl+Fuchs, “Incremental Rotary Encoder TVI40N”.[Online]. Available: https://files.pepperl-fuchs.com/webcat/navi/productInfo/edb/t39661_eng.pdf?v=20190314114807. [Accessed: 10-July-2019]

[10] Analog Devices, “ADXRS453 High Performance Digital Output Gyroscope”.[Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADXRS453.pdf. [Accessed: 10-July-2019]

[11] A.S. Conceicao, A. P. Moreira and P. J. Costa, ”Model Identification of a Four Wheeled Omni-Directional Mobile Robot”, Controlo 2006, 7th Portuguese Conference on Automatic Control, Instituto Superior T´ecnico, Lisboa, Portugal, 2006.

[12] M. H. Moradi, "New techniques for PID controller design," Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003., 2003, pp. 903-908 vol.2. doi: 10.1109/CCA.2003.1223130